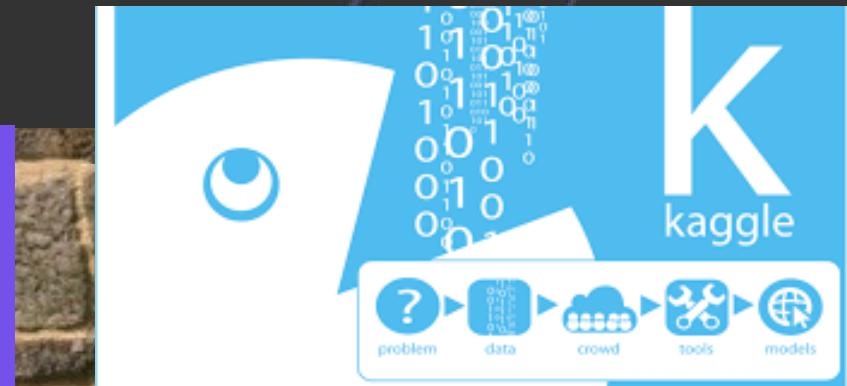


Kernel pour la competition Kaggle: Lyft Motion Prediction for autonomous Vehicles

OPENCLASSROOMS



Outline

- **Introduction**
 - Objectif de l'étude
 - Jeux de données
- **Analyse exploratoire**
 - Les données Lyft
 - Cartes semantiques
 - Coordonnées véhicules
- **Modélisation**
 - Modèle de départ
 - Pistes de modélisation
 - Optimisation
 - Contribution à la communauté Kaggle
- **Conclusions**
 - Résultat et possible amélioration

Modalités de la soutenance

5 min - Présentation de la problématique, de son interprétation et des pistes de recherche envisagées.

5 min - Présentation de l'exploration.

10 min - Présentation des différentes pistes de modélisation effectuées.

5 min - Présentation du modèle final sélectionné et résultats.

5 à 10 minutes de questions-réponses.

Objectif du modèle

Dans cette étude avec un apprentissage supervisé, on essaye de trouver le meilleur modèle pour

- Prédire la meilleure trajectoire pour un véhicule autonome
- Prédire le mouvement des autres véhicules détectés par la camera

Exploration des données

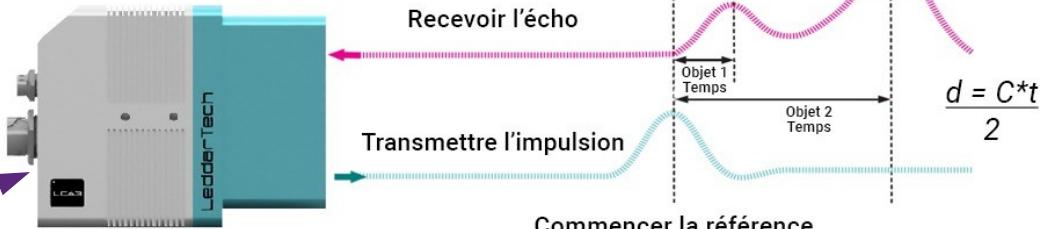
- Enregistrement
- Création des cartes sémantiques
- Jeux des entraînement, validation, test

Enregistrement des données

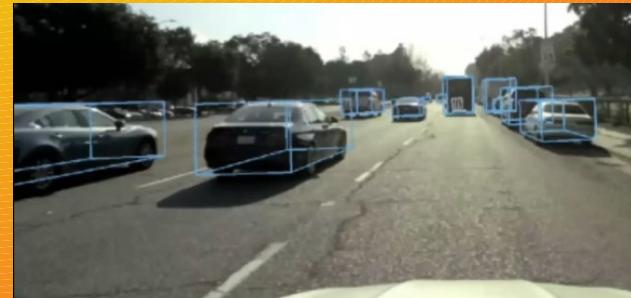


>1,000 hours
>3 billion tracked observations
23 AVs

Lidar : mesure la distance de chaque objet



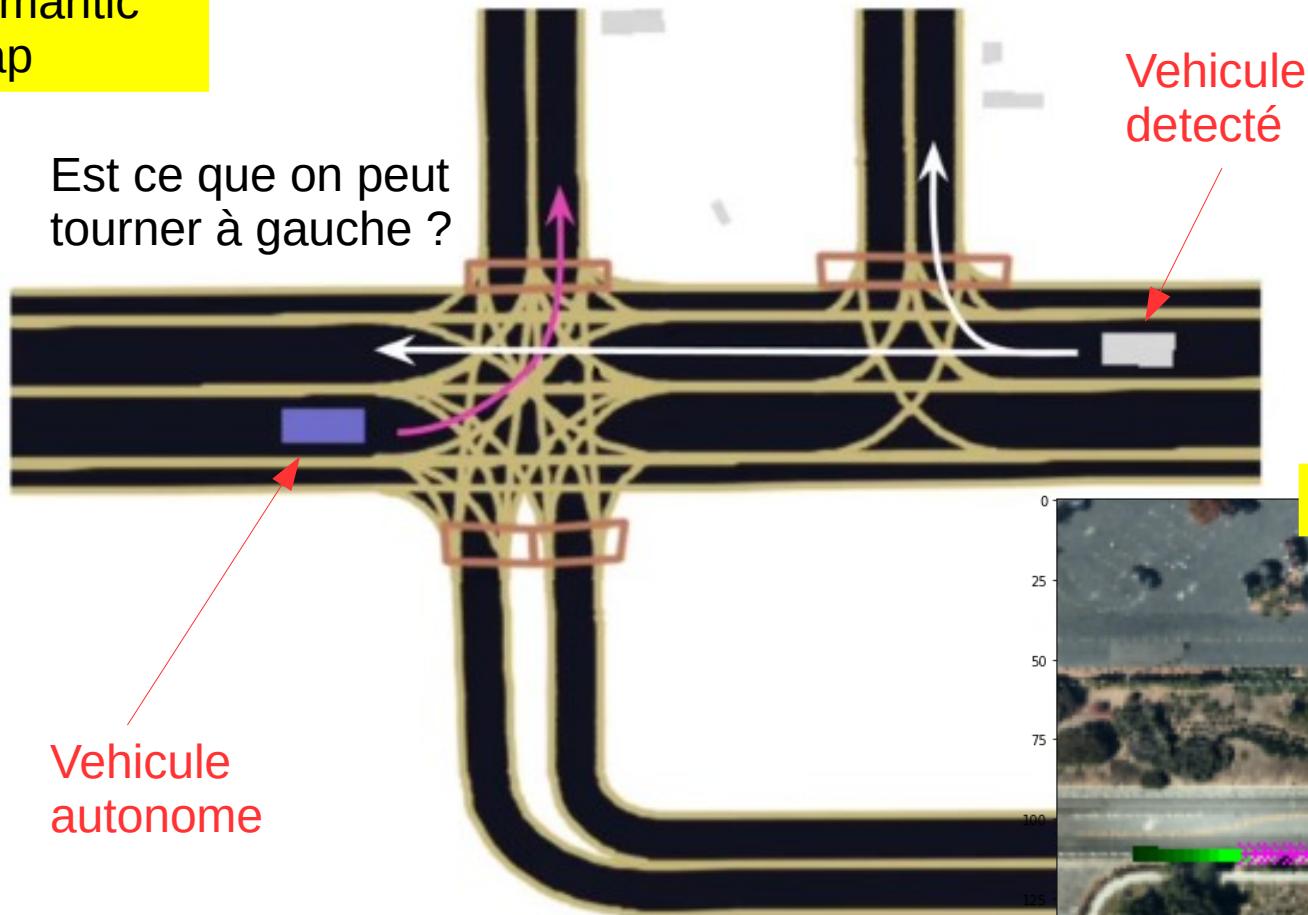
Perception et reconnaissance des objets (agents)



Integration avec les images satellitaires

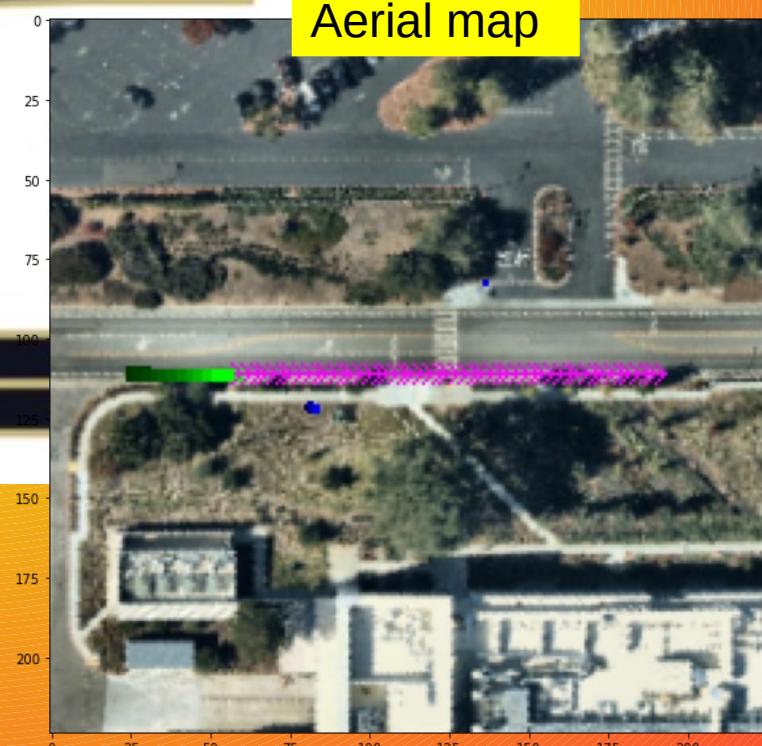
Semantic map

Est ce que on peut tourner à gauche ?



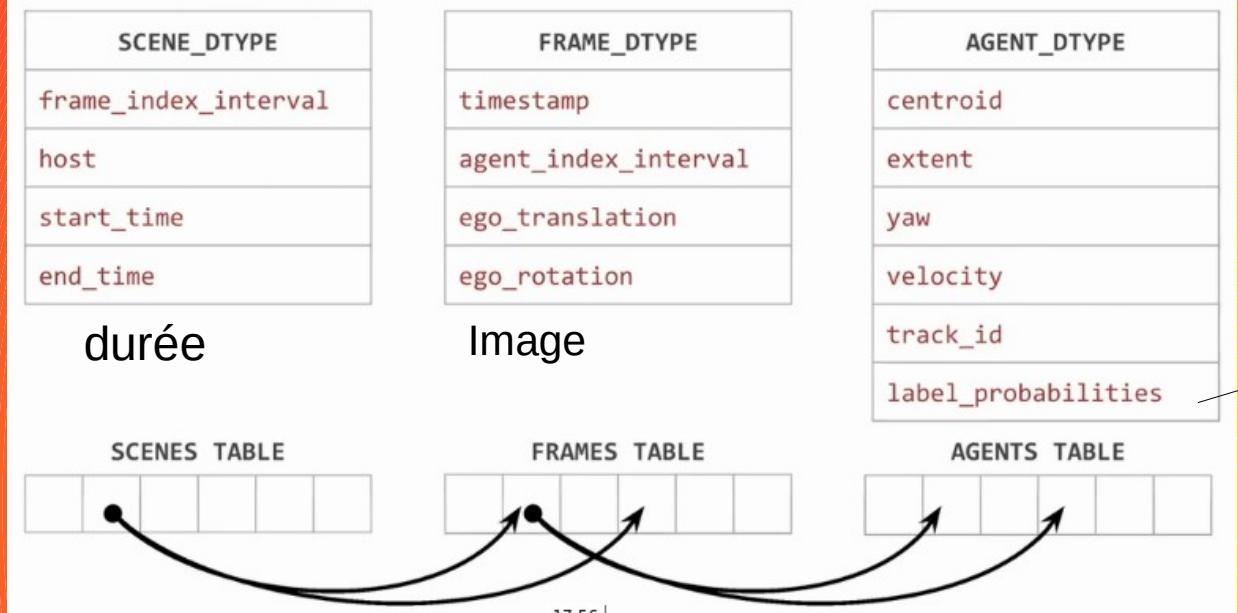
Véhicule détecté

Aerial map



Decision à prendre en fonction de l'apprentissage

Jeux des données



80 %
voiture
20 %
camion

Train : 134000 scenes

Validation : 11000 scenes

Test : 16000 scenes

Durée d'une scene 25s

250 frames X scene

Le format du dataset est optimisé pour PyTorch



Pour importer
les données
Lyft a créé des
librairies
spécialisées

L5KIT

Pytorch : avantages et désavantages



Vitesse de calcul
Mode « evaluation »
vs. « modélisation »

Complexité de code

Nécessité de créer
des fonctions 'class'
pour le modèle

Bugs d'évaluation

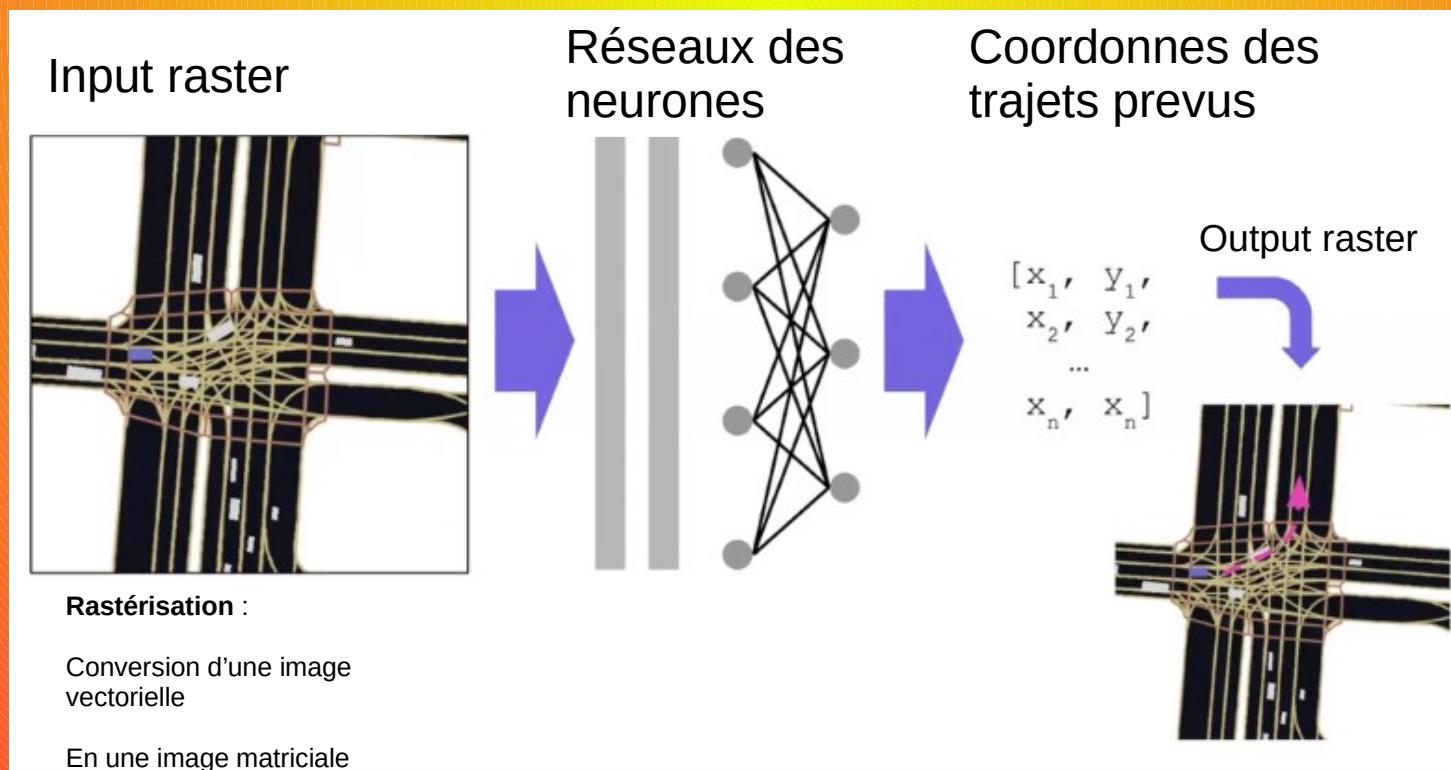
Facilité d'évaluation
des modèles

Visualisation directe
de la précision
(accuracy)

Modélisation

- Stratégie de prédiction
- Modèle initial
- Pistes de modélisation

Prédiction par Machine Learning



Modèle base

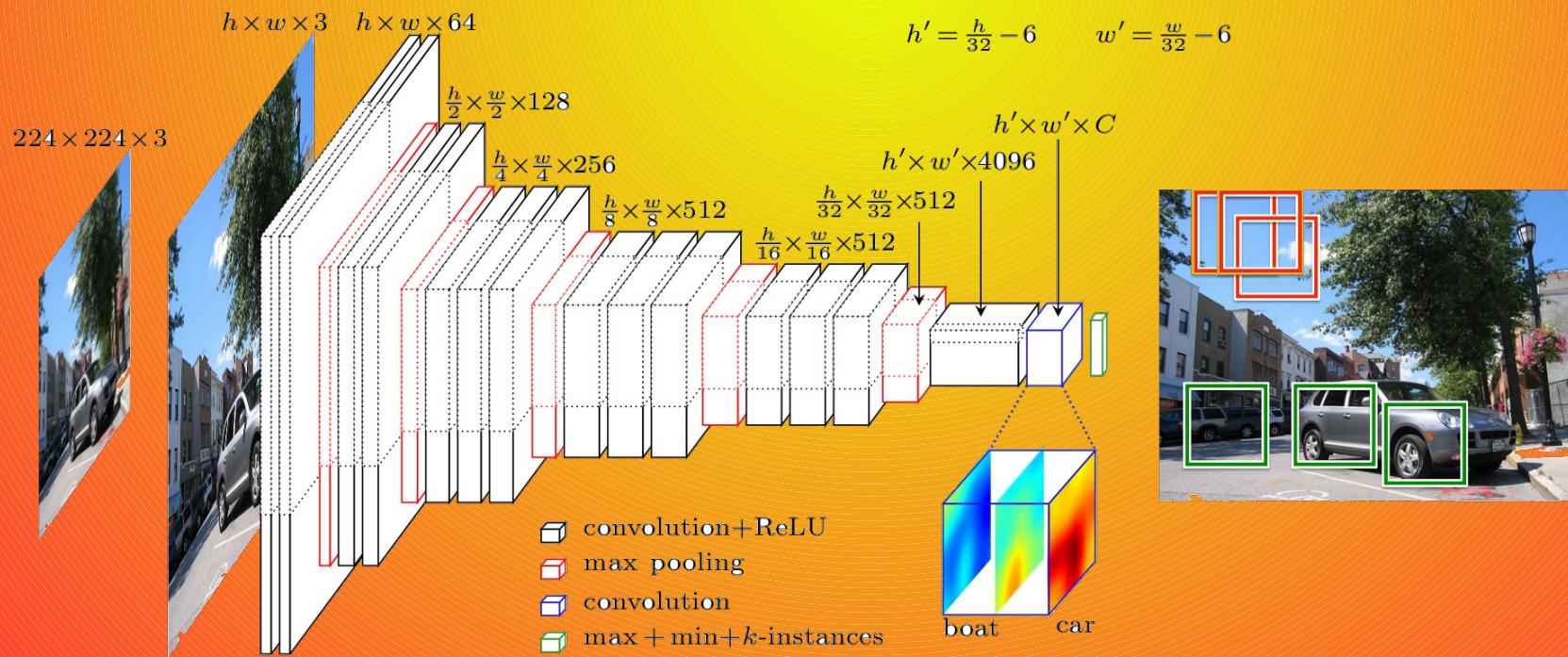
<https://www.kaggle.com/rhtsingh/gpu-training-lyft-torch-baseline>

https://github.com/lyft/l5kit/blob/master/examples/agent_motion_prediction/agent_motion_prediction.ipynb

- Resnet 50
- Criterion : MSELoss
- Optimizer : Adam
- Iterations (train) : 10000

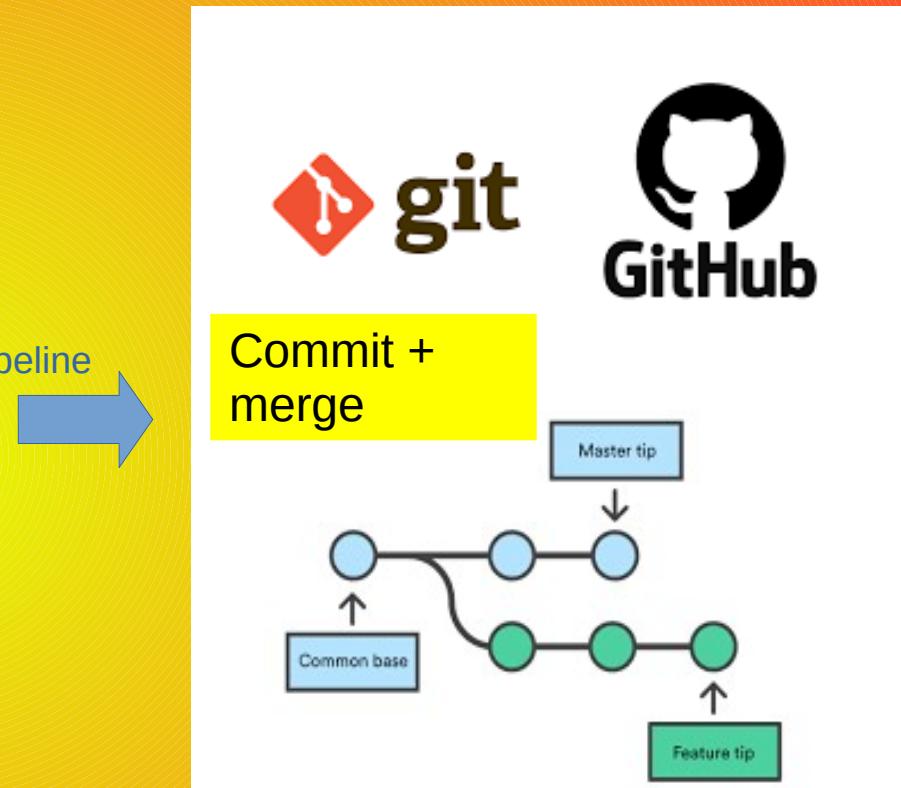


Public score : 1999



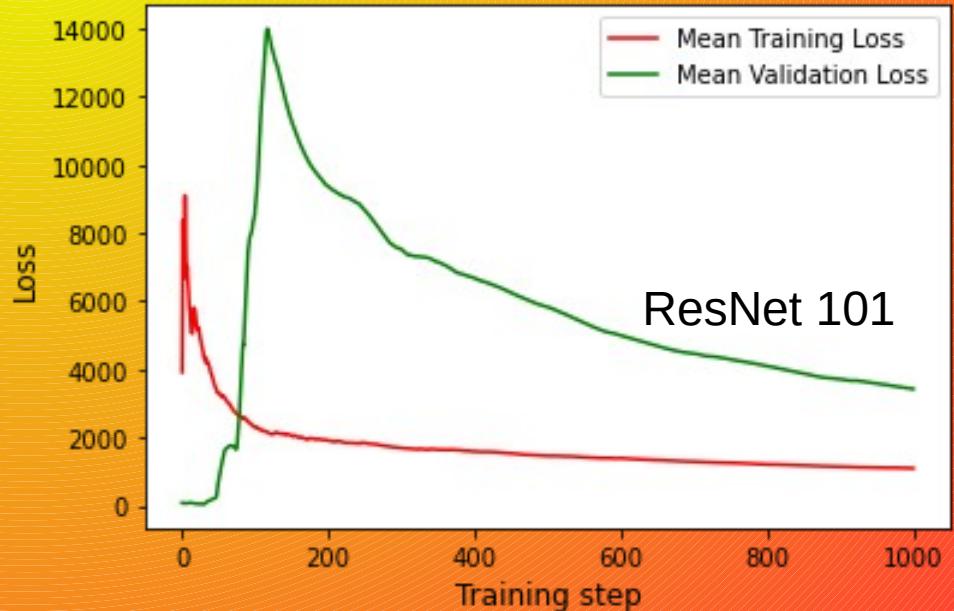
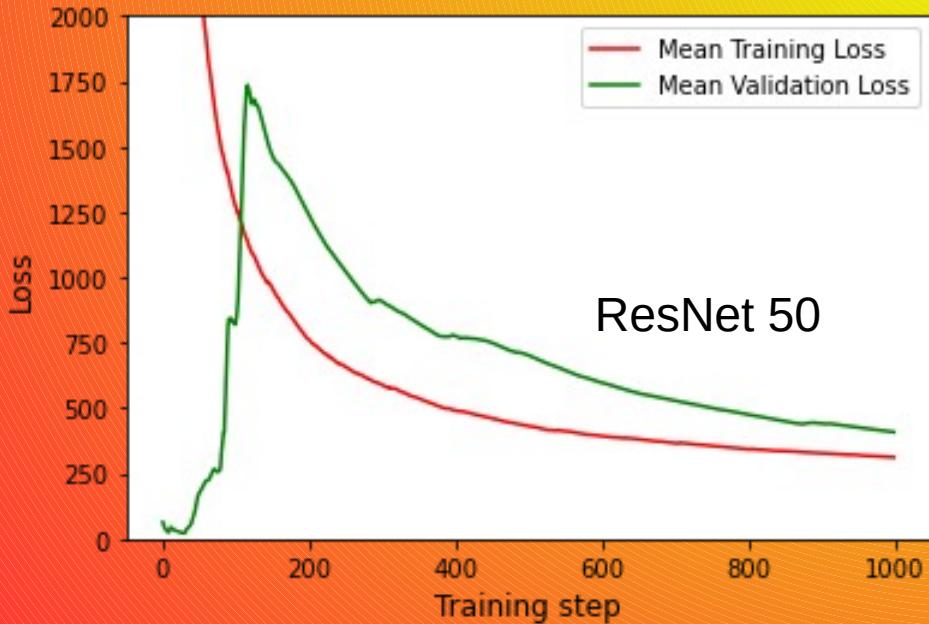
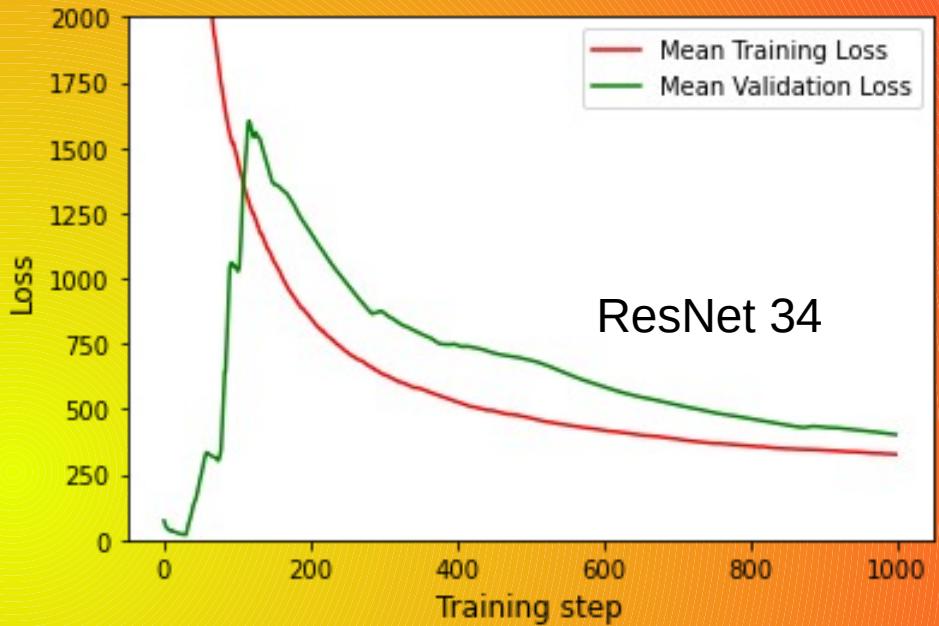
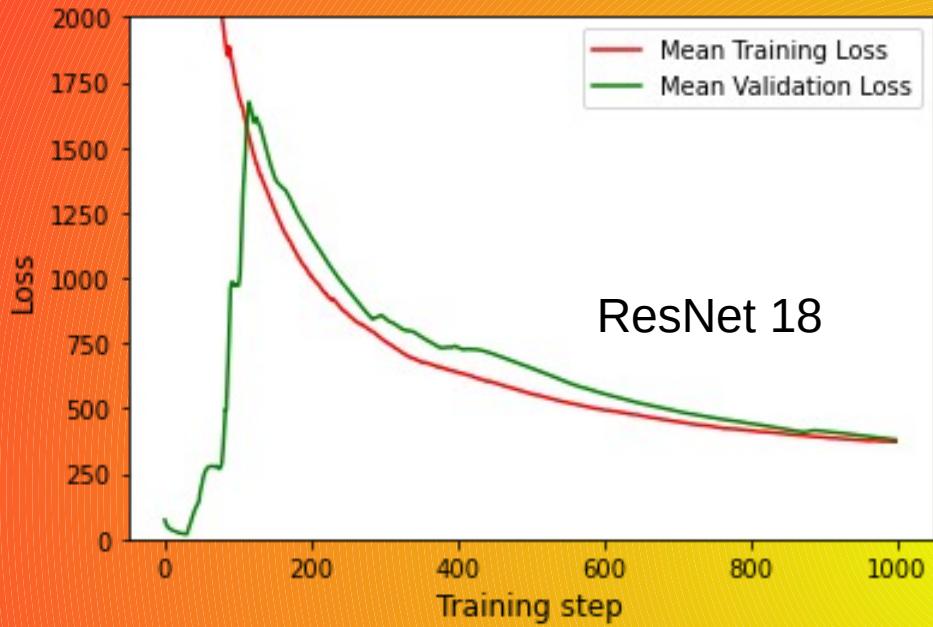
Modifications

- convergence avec optimisation SGD vs. Adam (pas efficace)
- optimisation : learning rate/batch size
- Non-Negative Likelyhood Loss vs MSELoss
<https://www.kaggle.com/huanvo/lyft-complete-train-and-prediction-pipeline>
- Changement d'architecture :
 - plus des couches denses pour ResNet
 - ResNet taille
 - Efficientnet



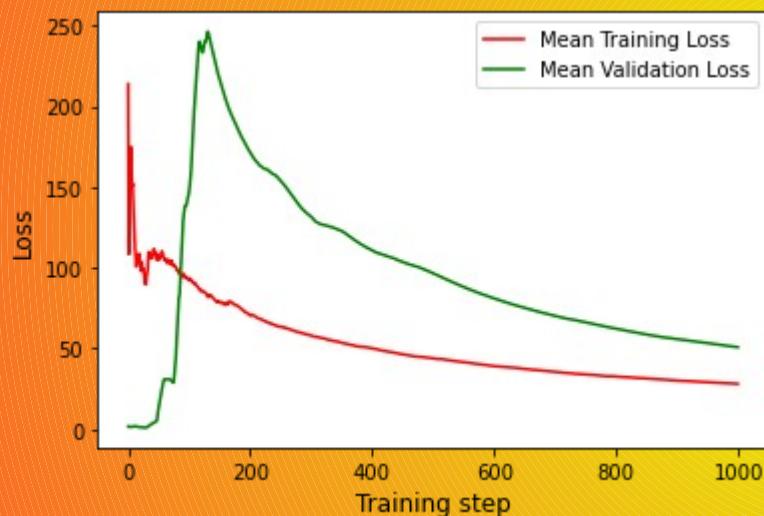
https://github.com/opsabarsec/Lyft_competition/

Architectures ResNet

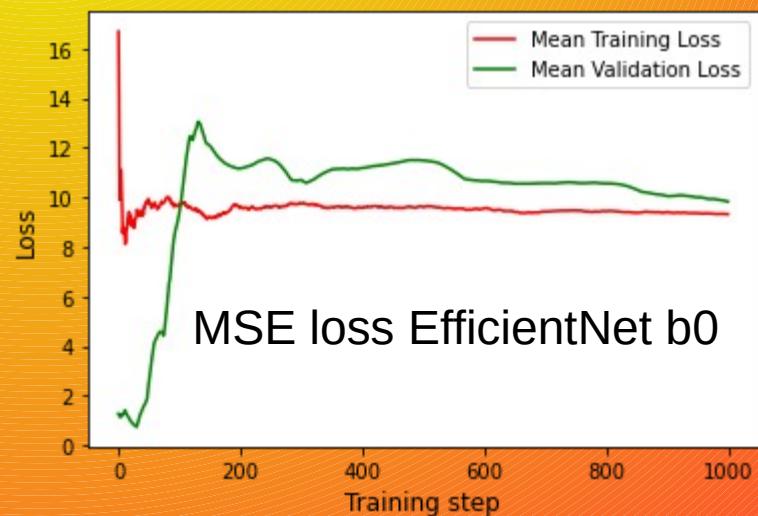
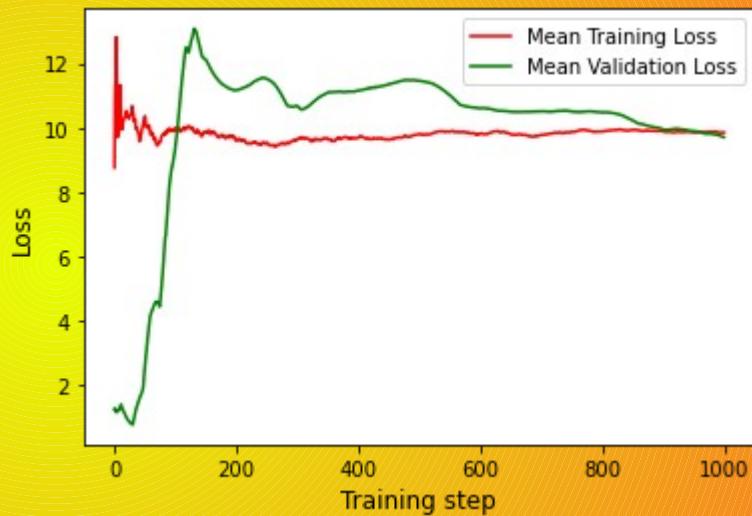


EfficientNet vs. ResNet

MSE loss ResNet 18

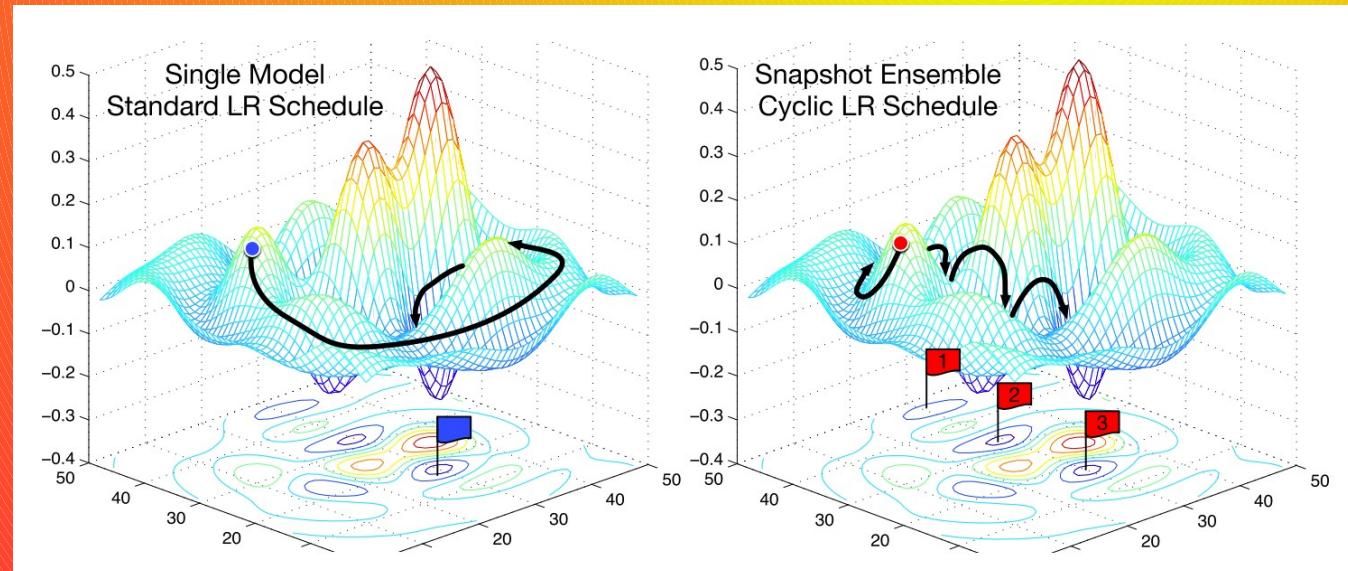
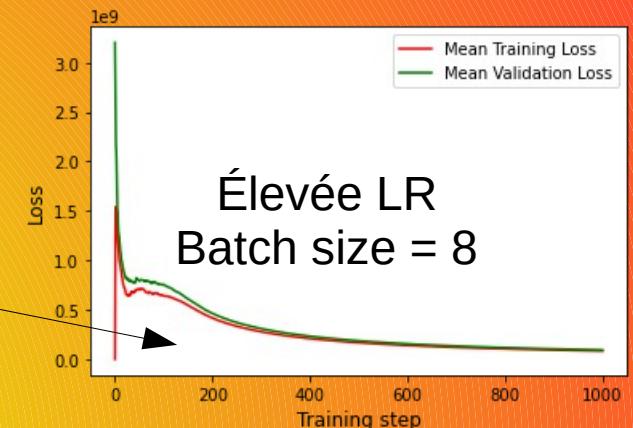
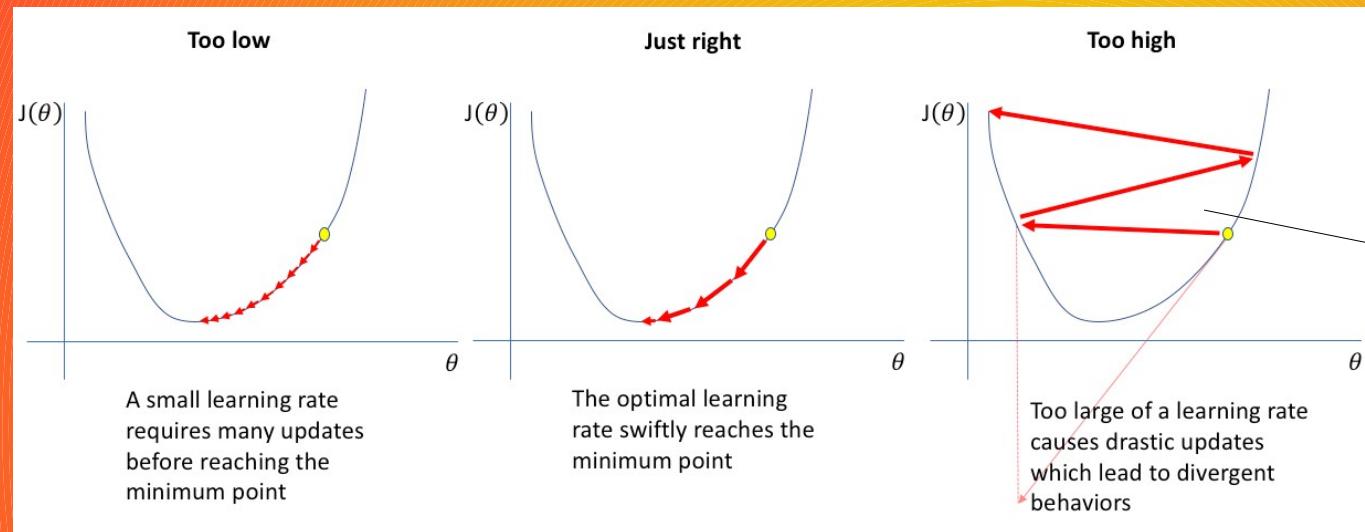


MSE loss EfficientNet b6



Optimisation de l'entraînement

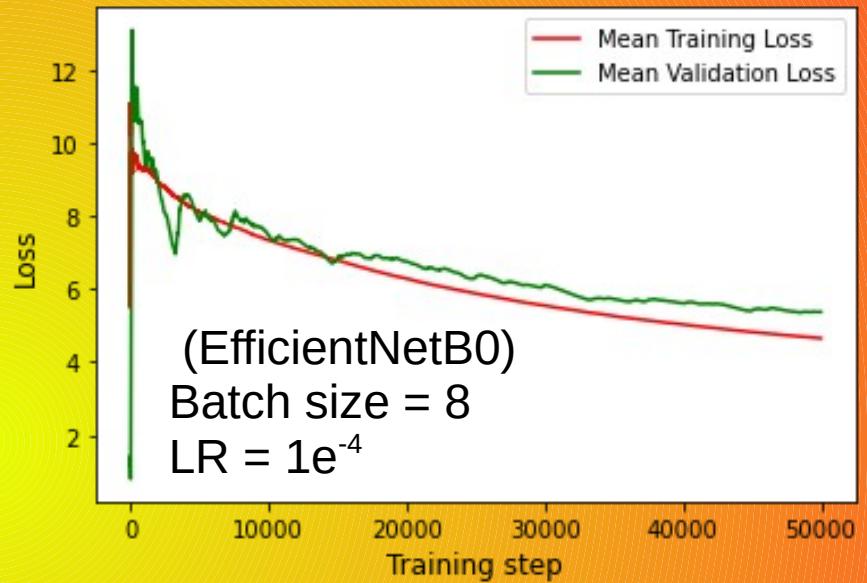
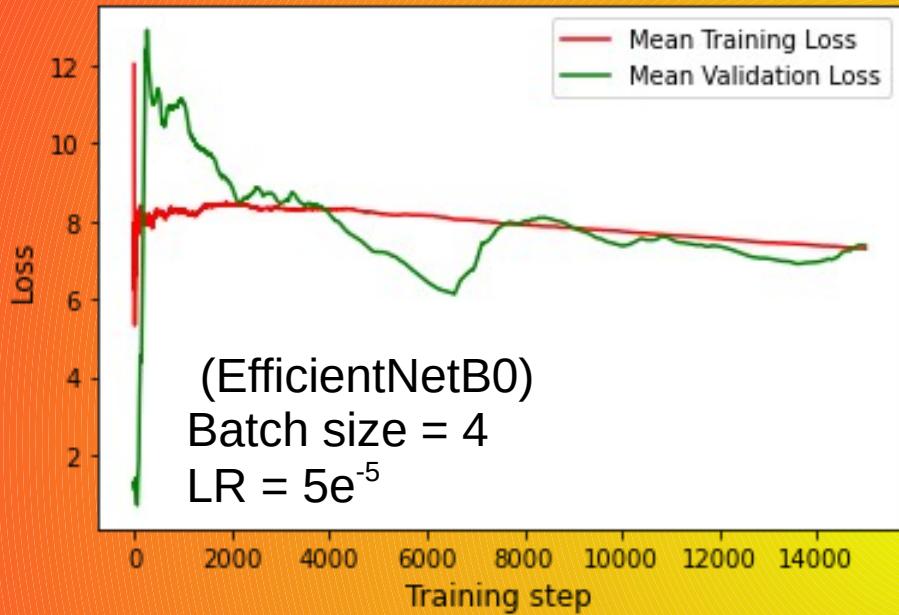
Learning rate



Batch size trop élevée risque de porter à un minimum local et empêcher la convergence du modèle*

* <https://mc.ai/5-concepts-you-must-know-to-pass-aws-machine-learning-certification-exam/>
<https://www.ijcai.org/Proceedings/2019/0604.pdf>

Performances and Limitations



Limitations

- RAM availability : même avec batch size = 4 et workers = 0 on est à la limite des 16GB donnés par Kaggle
- 30h GPU par semaine
- Instabilité du logiciel L5KIT

Contribution à la communauté Kaggle



Discussions Kaggle

- Intégration du logiciel propriétaire L5Kit

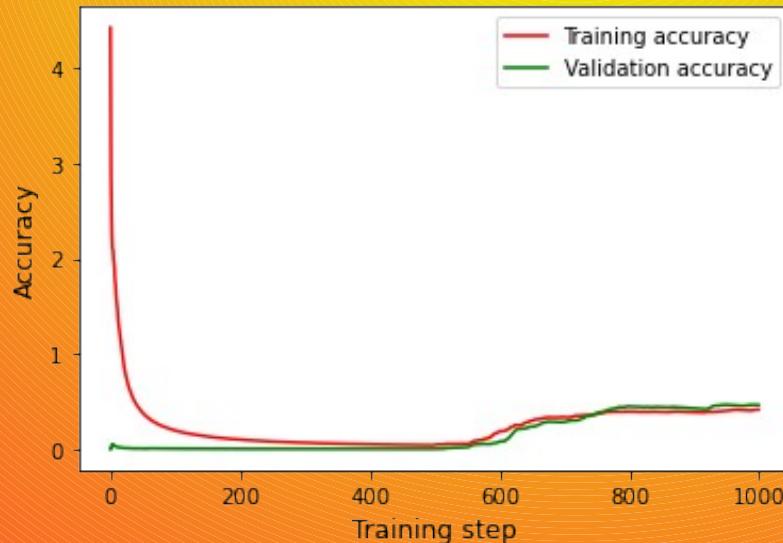
<https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles/discussion/186015>

- Possibilité d'utiliser la réseaux des neurones EfficientDet optimisée pour les cartes sémantiques

<https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles/discussion/185211>

- Précision (accuracy) et évaluation des prédictions

<https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles/discussion/187356>



Conclusions

- EfficientNet-B0 avec faible vitesse d'apprentissage (learning rate) donne la meilleure performance

Classement 1152 vs 1999 initial

- La Negative likelihood donne une meilleure estimation de l'erreur que le MeanSquareError
- Une possible amélioration des performances pourrait être obtenue avec :

Intégration TPU (pytorch lightning)

Définition des images (avec plus de RAM)
+ EfficientNet-b6

